Look-alike Modelling in Banking

Objective: Identify potential new customers who resemble existing high-value customers to target them with tailored marketing campaigns effectively.

Key Modelling Algorithm

1. **Random Forest:**
    - **Description:** Ensemble learning method that builds multiple decision trees and merges their predictions to improve accuracy.
    - **Use Case:** Effective for classifying potential customers as similar or dissimilar to high-value customers.

Important Variables for Look-alike Modelling (Indian Banking Perspective)

1. **Profiles of High-value Customers:**
    - **Attributes:** Transaction history, product holdings, creditworthiness, demographic information.
    - **Derived Attributes:** Lifetime value, profitability score, churn propensity.
2. **Transaction History of Potential Customers:**
    - **Attributes:** Transaction frequency, transaction amount, product usage.

Detailed Example of Look-alike Modelling Implementation

Step-by-Step Implementation Using Random Forest

1. **Data Preparation:**
    - Utilize existing campaign data (seed data) containing client attributes and campaign-related information.
    - Handle missing values and outliers, preprocess data for modelling.
2. **Feature Engineering:**
    - **Attributes:**

    - Transaction history
    - Product holdings
    - Creditworthiness
    - Demographic information
- **Derived Attributes:**
    - Lifetime value
    - Profitability score
    - Churn propensity

3. **Model Training and Evaluation:**

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from imblearn.over_sampling import SMOTE
from sklearn.metrics import accuracy_score, f1_score, roc_auc_score, precision_score

# Sample data preparation
# Assume df is the preprocessed and sampled data
X = df.drop('high_value_customer', axis=1)
y = df['high_value_customer']

# Split data into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Apply SMOTE to handle class imbalance
smote = SMOTE(random_state=42)
X_train_resampled, y_train_resampled = smote.fit_resample(X_train, y_train)

# Train Random Forest model
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train_resampled, y_train_resampled)
```

```
# Predictions on test data
y_pred = model.predict(X_test)

# Evaluation metrics
accuracy = accuracy_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)
roc_auc = roc_auc_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
```

4. **Interpretation and Actionable Insights:**
   - Analyze model performance using evaluation metrics like Accuracy, F1-Score, ROC-AUC, Precision.
   - Use the trained model to predict potential investors among the remaining client base (pool data).
   - Target the predicted look-alikes with personalized marketing campaigns to increase conversion rates and build a loyal and profitable customer base.

Variables for Targeting

1. **Look-alike Segment:**
   - **Attributes:** Transaction history, product holdings, creditworthiness, demographic information.
   - **Derived Attributes:** Lifetime value, profitability score, churn propensity.
   - **Target Variables:** Likelihood of becoming a high-value customer, expected lifetime value.

By following this approach, the bank can effectively identify and target potential customers who resemble their high-value customers, leading to improved marketing campaign effectiveness and overall business performance.